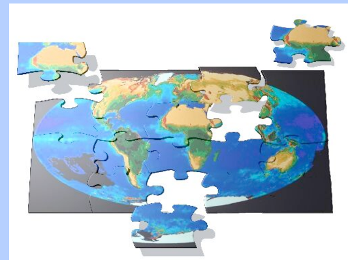
Ames Research Center

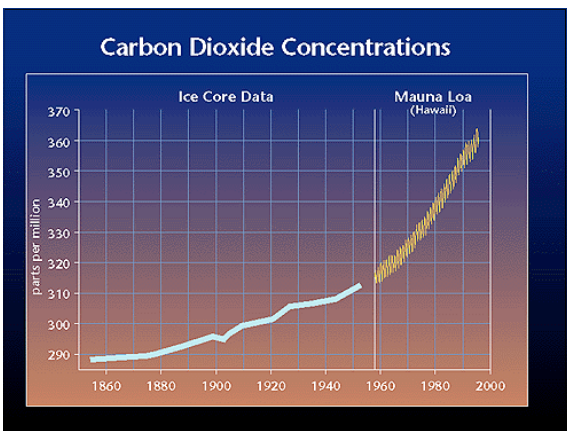
Intelligent Agents for Science Data Processing

Keith Golden,
NASA Ames

4/25/2003

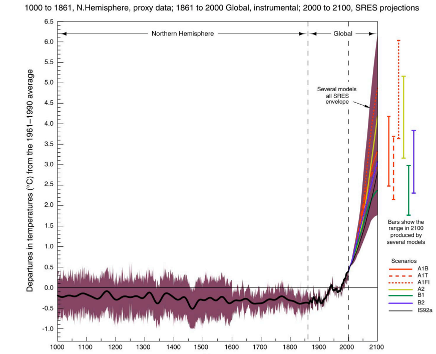


How is the Earth changing and what are the consequences for life on Earth?



Variations of the Earth's surface temperature 1000 to 2100

1000 to 1861, N.Hemisphere, proxy data; 1861 to 2000 Global, instrumental; 2000 to 2100, SRES projections



motivation

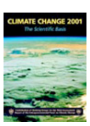


image credit: ipcc

Global environmental changes between 1980 and 2000 have been significant:

- Two of the warmest decades in the instrumental record
- Three intense El Nino events (1982-83; 1987-88; 1997-98)
- Changes in tropical cloudiness and monsoon dynamics
- A 9.3% increase in atmospheric co2 concentration
- A 36% increase in global population (4.45 billion in 1980 to 6.08 billion in 2000)

image credit: fao

A substantial incentive to understand trends and variability in terrestrial Net Primary Production because NPP:

- is the foundation of food, fiber and fuel for human consumption
- determines seasonal and interannual variations in atmospheric co2
- integrates climatic, ecological, geochemical and human influences on the biosphere

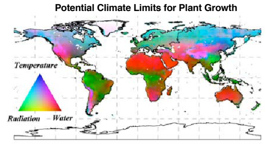
How have global environmental changes affected (eased or strengthened) climatic constraints to plant growth and NPP?

- the greening earth and increasing terrestrial npp, Science (forthcoming)

step 1: limiting factors

Plant growth is assumed to be principally limited by sub-optimal climatic conditions such as low temperatures, inadequate rainfall and cloudiness (Churkina and Running, 1998). We used 1960-1990 average climate data (Leemans and Cramer, 1991) to develop scaling factors between 0 and 1 that indicate the reduction in growth potential.

Potential Climate Limits for Plant Growth

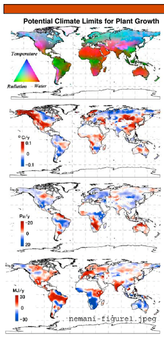


- black (no limits) and white (all at maximum limit)
- primary colors represent respective maximum limits
- cyan (temperate and radiation) represents cold winters and cloudy summers over eurasia
- magenta (water and temperature) represents cold winters and dry summers over western north america
- yellow (water and radiation) represents wet-cloudy and dry-hot periods induced by rainfall seasonality in the tropics
- these limits vary by season (e.g., high latitude regions are limited by temperature in the winter and by either water or radiation in the summer)

Dominant Controls
water availability 40%
temperature 33%
solar radiation 27%

- the greening earth and increasing terrestrial npp

step 2: trends in climate data



Data

Reanalysis data (6-hourly 2 m height temperatures, 2-m height specific humidity, and incident solar radiation) from the National Center for Environmental Prediction (NCEP) to represent climate variability from 1982 to 1999

Interannual trends in daily average temperature 1982-99

Spring air temperatures that regulate the initiation of the growing season have increased over temperature-limited regions of North America and Northwest Europe promoting earlier plant growth and additional carbon sequestration.

Interannual trend in vapor pressure deficit 1982-99

Wetter rainfall regimes and the associated reduction in vapor pressure deficits in the 1990s are important for the water-limited ecosystems of Australia, Africa and the Indian sub-continent.

Interannual trend in solar radiation 1982-99

Significant increases in incident solar radiation are evident over radiation-limited regions of Eurasia and the equatorial tropics. While the increased solar radiation over Eurasia may be related to changes in the North Atlantic Oscillation, a decline in cloud cover along with increases in outgoing long-wave radiation, as a consequence of changes in tropical circulation patterns, was recently reported over tropical regions.

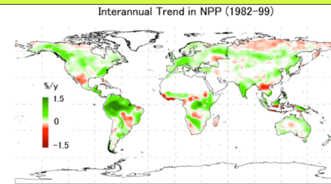
The observed climatic changes have been mostly in the direction of reducing climatic constraints to plant growth. Therefore, it seems likely that vegetation responded to such changes positively.

- the greening earth and increasing terrestrial npp

step 3: npp evaluation

The NPP Algorithm

Step 1 convert absorbed radiation to optimal gross production → **Step 2** downgrade by climate limiting factors to obtain gpp → **Step 3** subtract respiration to obtain npp

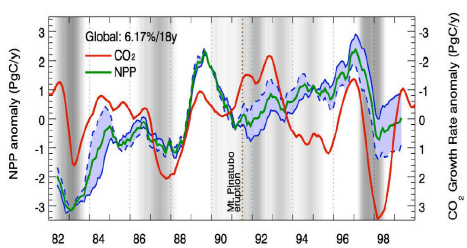


Average of interannual trends (1982-99) in growing season NPP estimated with GIMMS and PAL (v3) FPAR

Globally all biomes, except open-shrubs, showed an increasing NPP trend from 1982 to 1999 with the largest increase in evergreen broadleaf forests.
Trends in NPP are positive over 55% of the global vegetated area and are statistically more significant than the declining trends observed over 19% of the vegetated area.

- the greening earth and increasing terrestrial npp

climate, npp and atmospheric co2 growth rate



Although the atmospheric co2 growth rate depends on the net air-sea and land-atmosphere exchanges, these Results highlight the preeminent role of plant growth in global carbon cycle.

- the greening earth and increasing terrestrial npp

What it took:

3 research assistants for 12 months

processed <15 GB of data
9 global data sets for 20 years

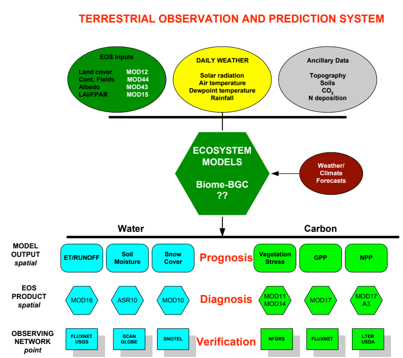
data analysis <20%

data preparation >80%

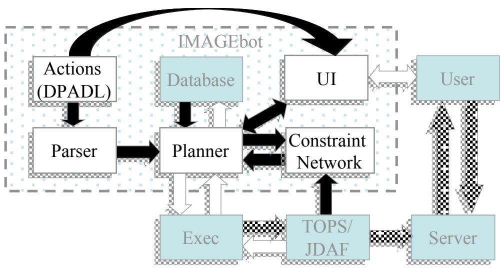
EOSDIS…

- **Generates ~3 tera-bytes of data a day.**
- **Currently holds 2 peta-bytes**
- **Grew 8 fold in volume since 1998.**
- **Continues growth ~2-3 tera-bytes a day**
- **Ingests 393 GB / day of raw data**
- **1 day = 2 years of HUBBLE Space Telescope**
- **1 day = more than 3 years of UARS**

Synergistic analysis of EOS products



Agent architecture



Choices & Tradeoffs

- Input data
  - Spatial resolution and extent
  - Temporal resolution and extent
  - Availability, timeliness
  - Quality
    - Sensor reliability / clouds
    - Historical / simulated
- Models
  - Input requirements
  - Time, resources
  - Output quality

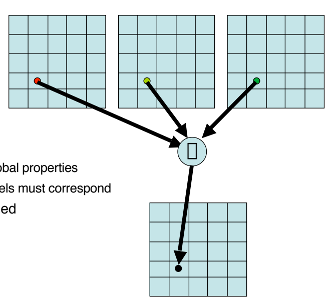
Input data choices

Terra-MODIS	FPAR/LAI	1 day	1km, 500m, 250m	global, since 2000
Aqua-MODIS	FPAR/LAI	1 day	1km, 500m, 250m	global, since 2002
AVHRR	FPAR/LAI	10 day	1 km	global, since 1981
SeaWIFS	FPAR/LAI	1 day	1xm x 4 km	global
DAO	temp, precip, rad, humid	1 day	1.25 deg x 1.0 deg	global, since 1980
RUC2	temp, precip, rad, humid	1 hour	40 km	USA
CPC	temp, precipication	1 day	point data	USA
Snotel	temp, precipitation	1 day	point data	USA
GCIP	radiation	1 day	0.5 deg	Continental
NEXRAD	precipitation	1 day	4 km	USA

- Imagebot
  - Action rep, KR
  - Planning
    - Optimization
    - Regression, lifted plan graphs, graph expansion
  - Constraint reasoning
    - Forall, strings, attachments
  - IPE/Sensing

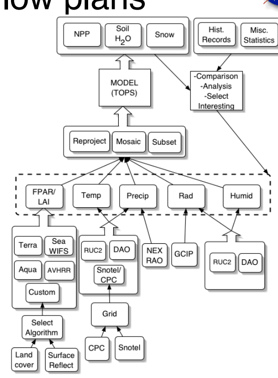
Feeding the model

- Pixel-based operations
- Inputs must agree on
  - projection
  - resolution
  - region
  - time
- Model is "myopic"
  - doesn't care about global properties
  - but corresponding pixels must correspond
- Inputs can be transformed
  - reprojection
  - scaling
  - mosaic
  - cropping
- Choose best inputs given constraints and preferences



Dataflow plans

- Data sources
  - Images, tables, semi-structured text
  - Described in terms of
    - Information content
    - Data format
    - Quality, resolution, …
- Filters
  - low-level data transformation
    - described at level of data
    - mosaic, subset, reproject, …
- Models
  - high-level transformation
    - convert data to information
    - NPP, soil moisture, fire risk, …
  - convert data to information



Why not use relational DB?

- Input data are stored in binary files
  - Terabytes or more in concise format
- Output data are stored in binary files
  - Users want images, not DB tables
  - Standard formats used by scientists
- Programs optimized for binary data
  - Image processing procedures
  - Mosaic, reproject, etc.
  - Models assume grid input/output

Desired "programming language" properties

- Naturally describe domain concepts
  - Data semantics and syntax
  - Structured data files, complex data types
  - Object creation/copying
- Specify interfaces external environment
  - Execute plans
  - Obtain information (sensing)
- Easily usable by programmers
  - Similarity to known programming languages
  - As feature-rich as necessary

Desired "planning language" properties

- Easily useable by planners
  - Declarative semantics
  - States, actions
  - As feature-poor as possible
- Compatible objectives?
  - Commonality: state, state change
    - Variables
    - Actions ≈ Procedures/methods
    - Preconditions ≈ tests
    - Conditional effects ≈ conditional instructions
  - "Compile" to simpler language

DPADL: Data Processing Action Description Language

- Object oriented, C++/Java-like syntax
  - Inheritance
  - Primitive types and objects
  - Object creation, copying, modification
- Integration with Java
  - Embedded Java code
    - Action execution
    - "Procedural" constraints
  - Parameters include Java objects
- Actions describe data-processing operations
  - Any number of inputs and outputs
  - Causal, declarative representation of data filters
- Constraints over any static type

Types

- Can inherit from objects or primitive types

```
static type Filename instanceof String;
```

- Can be defined by list of members

```
static type ImageFormat =
    {"JPG", "GIF", "TIFF", "PNG", "XCF", … };
static type ProjectionType =
    {LAZEA=11, GOODE_HOMOL=24, ROBINSON=21, …
  };
```

- Can represent complex data structures

```
type Image instanceof Object {
    static int xSize;
    static int ySize;
    PixelValue pixelValue(int x, int y);
    …
```


Functions

- Can be static or fluent

```
fluent float temperature(real lon, real lat);
static float sin(real x);
```

- Are the atoms of the language
  - Attributes are functions of their objects
  - Infix operators are functions (C++ style overloading)

    ```
    static String operator+(string s1, string s2);
    ```
  - Relations are boolean functions

    ```
    static boolean operator<(real r1, real r2);
    ```
  - Global variables are fluent functions with no arguments

    ```
    fluent Date currentDate;
    ```
- Can be targets of assignment

```
image1.pixelValue(x, y) := image2.pixelValue(y,
  x);
```

Constraints

Can be attached to types or functions

```
static type Filename isa string {
    constraint Matches(true, this, "~[/]+");
}
static string operator+ (string s1, string s2) {
    constraint Concat(value, s1, s2);
}
```

Can be specified using embedded Java code.

```
static type Tile isa object mapsto
  tops.modis.Tile {
   Instrument product {
    constraint {
      result(this) := $ this.getProduct() $;
    }
```


Constraints

```
static type Tile isa object mapsto tops.modis.Tile
  { …
  //true if this tile covers the specified location
  boolean covers(real lon, real lat) {
    constraint {
      {this}([lon], [lat], d=day, y=year,
             p=product, value)
        := {$ if(value)
              return tm.getTiles(lon.max, lat.min,
                                 lon.min, lat.max,
                                 d, y, p);
              else return null; $ };
    }
…
```


Actions

```
action threshold (unsigned thresh) {
  input BwImage in;
  output BwImage out copyof in;
  forall unsigned x, unsigned y;
  effect when (x < in.xSize && y < in.ySize) {
              when (in.valueAt(x, y) <= thresh)
  {
                 out.valueAt(x, y) := BLACK;
              } else {
                 out.valueAt(x, y) := WHITE;
              }
           }
  exec $ out = gfx.threshold(in, thresh); $;
```


Summary

- Data processing is an interesting application for planning
- Writing domain descriptions is programming
- Language should provide support for data domain concepts
  - Complex data types
  - Constraints
  - Integration with runtime system

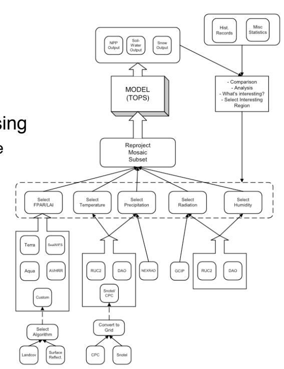
Data Goals and Metadata

What
How
Where
When

- Data product specification
  - **What** information is contained
  - **How** information is encoded in data
  - **Where** the data files are stored/delivered
  - **When** the information pertains to
- Examples
  - I want an MPEG movie of yesterday's weather over the SF bay placed on our website
  - File dd010101.tar.gz is a compressed archive of the downlink directory as of Jan 1, 2001

Planning

- Planning for data processing
  - Initial state = data available
  - Goal = data to produce
  - Plan = dataflow program
- Domain characteristics
  - Very large universes
  - Large plans
  - Lots of constraints
  - \+ Highly parallel



Representational Challenges

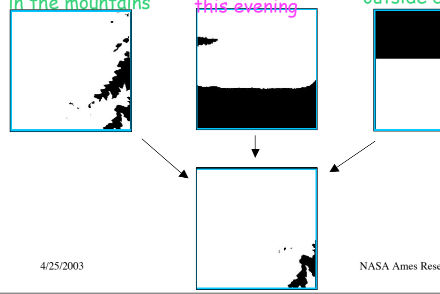
- Universal quantification in actions and goals
  - All files in the directory
  - All pixels in the image
- Incomplete information
  - Files on internet
- Infinite universes
  - Real-valued quantities
  - Filenames, data contents
- Constraints

The picky stargazer

Find me a location... in the mountains

with no cloud cover ~~this evening~~

outside city limits



What
How
Where
When

Image transformations

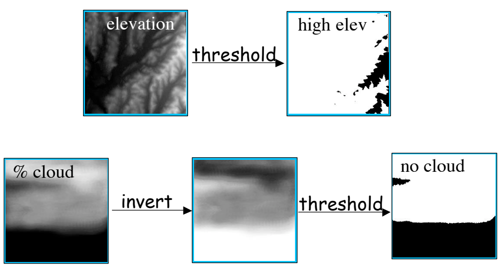


Image transformations

Threshold(*th*): (input = in, output = out)
□*x*. **when**(*in*.pixel-value(*x*, *y*) □ *th*)
out.bitValue(*x*, *y*) := 0)
else
out.bitValue(*x*, *y*) := 1)

Invert: (input = in, output = out)
□*x*, *y*. *out*.pixelValue(*x*, *y*) := 255 - *in*.pixelValue(*x*, *y*)

Elevation map:
when(0 □ *x*' < 500 && 0 □ *y*' < 500
&& proj(x',y',lon',lat')
&& elev = elevation(lon', lat'))
image.pixelValue(*x*', *y*') = f(e)

What
How
Where
When

Planning for Universally quantified goals

- Traditional approach: universal base
  - **original goal** □*x* : block. onTable(*x*)
  - **new goal** onTable(A); onTable(B); onTable(C)
- May be impossible or impractical to enumerate all objects in universe
  - "delete all files on disk"
- Enumerating universe not always required
- Satisfy □ goals directly with □ effects
  - shake-table: □*x* : block. onTable(*x*)
  - rm -rf /
  - lpr papers/*.ps
- Supported by PUCCINI planner

Subgoaling

PS(a)

pdf2ps *.pdf
□*z* **when**(PDF(*z*)) *PS*(*z*)

lpr *.ps
□*x* **when**(PS(*x*)) *Pr*(*x*)

Pr(a)

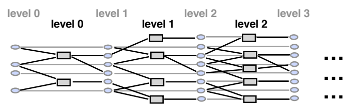
□*y* **when**(PDF(*y*)) *PDF*(*y*)

□*y* **when**(PDF(*y*)) *PS*(*y*)

□*y* **when**(PDF(*y*)) *Pr*(*y*)

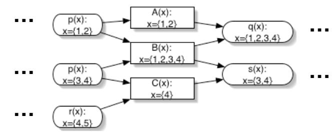
Trivially satisfied

Plan graphs



- \+ Good data structure for reachability/cost analysis
  - \+ Polynomial in problem size
  - \+ Cost = number of steps until proposition(s) possible
- – Requires grounded representation
  - – All possible ground actions
  - – All possible ground propositions
  - – Not feasible when number of objects is large

Lifted plan graph



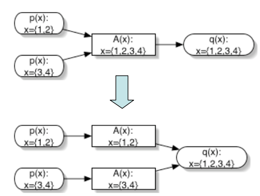
- \+ Graph size independent of number of objects
- \+ Efficient constraint propagation through graph
- – Less informative than grounded graph
  - – No mutexes (hard when there are variables)
  - – Union of variable domains conceals inconsistencies

Graph expansion

- Expand graph to expose inconsistent assignments
  - Split nodes with inconsistent producers or consumers
- Exposes conflicts for
  - Domain reduction
  - Node elimination
  - Better distance estimates
  - More opportunities for splitting

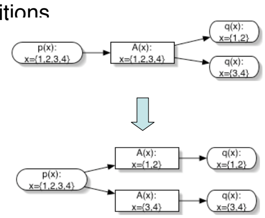
Forward expansion

Expand nodes with inconsistent supporters



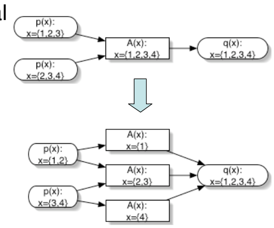
Backward expansion

Expand nodes supporting inconsistent conditions



Expansion to force conflicts

Expand when domains are consistent but not equal



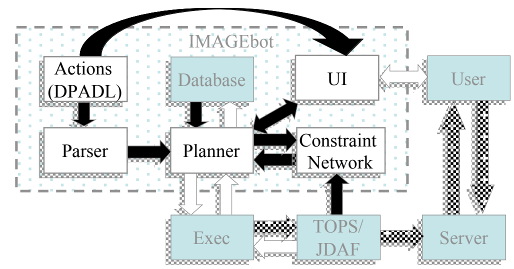
Issues

- Expansions apply recursively – when to stop?
  - + More accurate distance heuristics
  - + Tighter bounds on variables for constraint reasoning
  - – Larger graph, constraint network
- Variable domains strictly decrease, so expansion
  - + Can be viewed as part of propagation
  - + Is not a backtrack point

– Graph represents whole search space

+ Focus on subgraph with lowest heuristic cost

Agent architecture



Quantified constraints

- Subgoal *trivially satisfied* if RHS entailed by LHS
  - □ *y* **when**(PDF(*y*)) PDF(*y*)
  - PDF(y) entails PDF(y) (determined by matching)
- What if R is a constraint?
  - □ *x* **when**(*x=0*) *x<1*
  - □ *n* **when**(*n*.matches(".*\.ps")) *n*.contains("ps")
- Universally quantified constraints

Quantified constraint semantics

$$\Box \vec{x},\vec{y}\ \Box\vec{z}\ A(\vec{x},\vec{y})\ \Box\ B(\vec{x},\vec{z})$$

$$\{\vec{x}:A(\vec{x},\vec{y})\}\ \Box\ \{\vec{x}:B(\vec{x},\vec{z})\}$$

Quantified constraint example

$$\Box\ x,y\Box\ \mathbb{N}\ \Box z\Box\ \mathbb{N} > x+y=3\ \Box\ x<4$$

$$\{x,y\} = \{(1,2),(2,1)\}$$

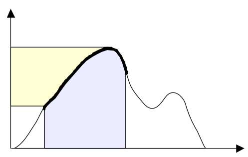
$$\{1,2\}\ \Box\ \{1,2,3\}$$

Infinite domains

- What if the domain of *x* is infinite?
  - How to identify all *x* that satisfy constraints?
  - Not always possible
- However, if
  - domain has concise representation
  - all constraints preserve concise representation
- then it may be possible to store the exact values satisfying A(*x*,*y*), B(*x*,*z*) in the domain of *x*.

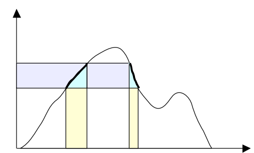
Numeric domains

- Intervals provide concise representation
- If domain of continuous function is interval, so is range



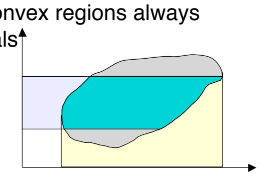
Numeric domains

- Intervals provide concise representation
- If domain of continuous function is interval, so is range, but converse is false



Numeric domains

- Intervals provide concise representation
- If domain of continuous function is interval, so is range, but converse is false
- Relations describing convex regions always map intervals to intervals
  E.g
  □ $x > 5$
  □ $x < 2y$
  □ $x^2 + y^2 < 16$



String domains

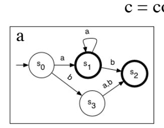
- Regular expressions provide concise representation
- Far more expressive than intervals
  - Ordinary string constraints easily captured
  - Closed under intersection, union, negation
  - Also more expensive

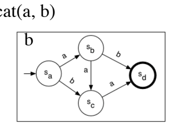
String constraints

- matches(string, regexp)
- c = concat(a, b)
- length(string) = n
- charAt(string, num) = char
- contains(string, substring)
- substitute(string, char1, char2)

String concatenation

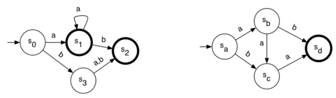
c = concat(a, b)





String concatenation

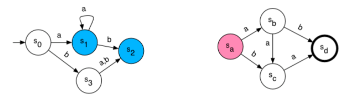
c = concat(a, b)



4/25/2003 NASA Ames Research Center

String concatenation

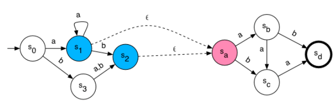
c = concat(a, b)



4/25/2003 NASA Ames Research Center

String concatenation

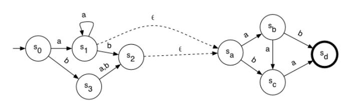
c = concat(a, b)



4/25/2003 NASA Ames Research Center

String concatenation

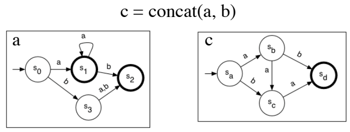
c = concat(a, b)



4/25/2003 NASA Ames Research Center

Inverse concatenation

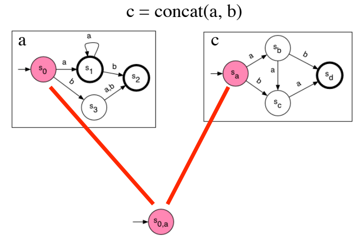
c = concat(a, b)



4/25/2003 NASA Ames Research Center

Inverse concatenation

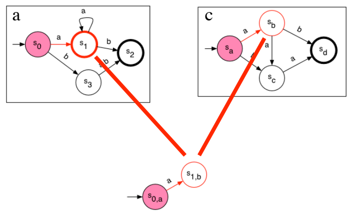
c = concat(a, b)



4/25/2003 NASA Ames Research Center

Inverse concatenation

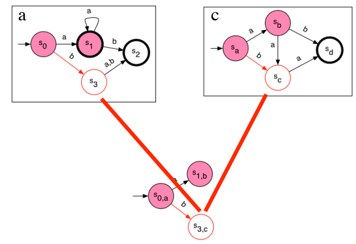
c = concat(a, b)



4/25/2003 NASA Ames Research Center

Inverse concatenation

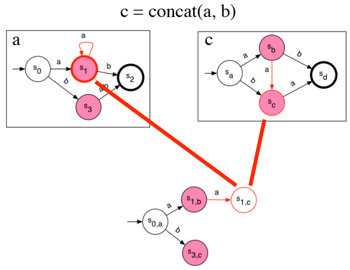
c = concat(a, b)



4/25/2003 NASA Ames Research Center

Inverse concatenation

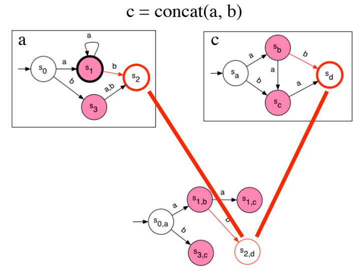
c = concat(a, b)



4/25/2003 NASA Ames Research Center

Inverse concatenation

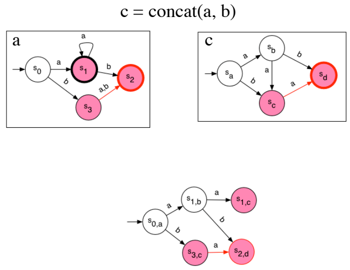
c = concat(a, b)



4/25/2003 NASA Ames Research Center

Inverse concatenation

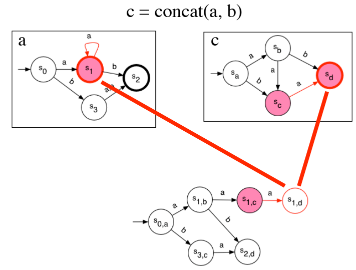
c = concat(a, b)



4/25/2003 NASA Ames Research Center

Inverse concatenation

c = concat(a, b)

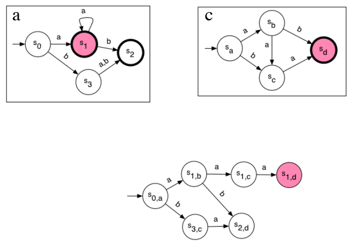


4/25/2003 NASA Ames Research Center

Ames Research Center

Inverse concatenation

c = concat(a, b)

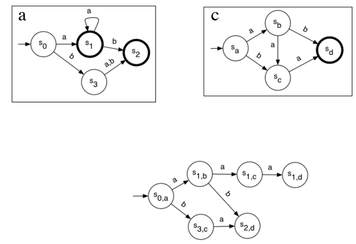


4/25/2003 NASA Ames Research Center

Ames Research Center

Inverse concatenation

c = concat(a, b)

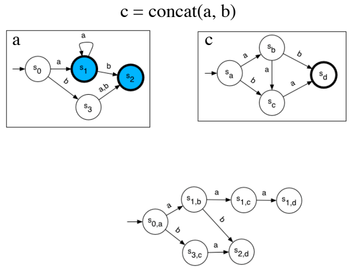


4/25/2003 NASA Ames Research Center

Ames Research Center

Inverse concatenation

c = concat(a, b)

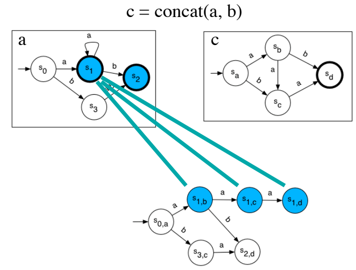


4/25/2003 NASA Ames Research Center

Ames Research Center

Inverse concatenation

c = concat(a, b)

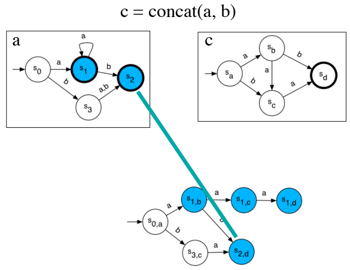


4/25/2003 NASA Ames Research Center

Ames Research Center

Inverse concatenation

c = concat(a, b)

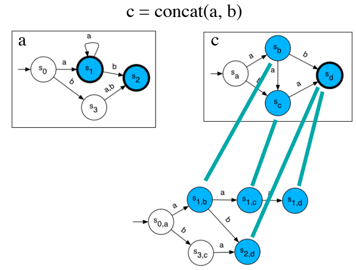


4/25/2003 NASA Ames Research Center

Ames Research Center

Inverse concatenation

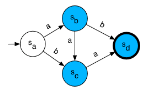
c = concat(a, b)



4/25/2003 NASA Ames Research Center

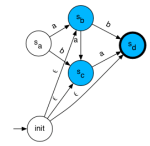
Inverse concatenation

c = concat(a, b)



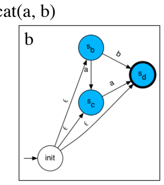
Inverse concatenation

c = concat(a, b)



Inverse concatenation

c = concat(a, b)



Some definitions (Freuder)

- **k-consistent**: any consistent assignment to k-1 variables can be extended to a consistent assignment to k variables
- **strongly k-consistent**: k-consistent for all j <= k

arc consistent: k-consistent for

k=1

a=b

{1 ~~2~~ 3} {1 ~~4~~}

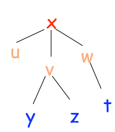
Some definitions (Freuder)

- Given: CSP □, variable x, and variable ordering O
  – width(x) in O = # variables before x in O sharing a constraint with x
  – width(O) = $\max_x$(width(x))
  – width(□) = $\min_O$(width(O))

E.g., tree-structured CSP

O = y,v,w,t,z,u,x
width(□) = width(x) = 3

O = x,u,v,w,y,z,t
width(□) = 1



Theorem

- Given the universally quantified constraint
  □x,$\vec{y}$ □$\vec{z}$ □(x,$\vec{y}$) □ □(x,$\vec{z}$)
- If
  □ and □ share one □ variable x
  □ and □ are strongly k-consistent
  – x is the first variable in an ordering that induces width < k for □ and □
- Then the constraint is satisfied iff $d_□$(x) □ $d_□$(x)

Ames Research Center

Current and Future work

- Current application: Terrestrial Observation and Prediction System (TOPS)
  - Planner integrated with TOPS
  - Solves simple problems, but work in progress
- Forall constraints involving two or more shared variables
- Finding optimal or high-quality plans
  - Utility = numeric function over data quality, time, resources
- Probability

Ames Research Center

Related Work

- ACWG
  - Distributed workflow generation for grid.
  - No causal reasoning about data
  - Optimization of time, resources, but not quality
- MVP, COLLAGE
  - Scientific image processing, human in the loop
  - HTN representation, less need for precise causal representation
  - No causal reasoning about data, metadata generation
- Chimera
  - Data tracking, but no support for causal reasoning
- Amphion, AutoBayes
  - Program synthesis using theorem proving
  - More expressiveness than needed for many DP problems
- Information Integration & Web Services
  - Information, not data